

Python Summary 5: Loops 2:

While loops

Loops:

Loops allow us to repeat an action some number of times. In Python, there are two main types of loops that are used: for loops and while loops.

for loops iterate over (go through each item in turn) a list, and stop when they reach the end of the list. *while* loops contain a condition, and repeat until the condition is False.

While loops:

A while loop starts with `while <condition>:`. The code inside the while loop will be run repeatedly until the condition is False. The condition is checked *before* each time the code in the loop is run. What does the following loop do? Run it in IDLE to check.

Example:

```
num = 0
while num < 10:
    print(num)
    num = num + 1
```

When creating a while loop, you need to be careful that you don't end up with an infinite loop. If the condition of the while loop is always True, then the loop will continue running forever. What does the following loop do if you run it in IDLE?

```
num = 0
while num < 10:
    print(num)
```

What is different about the second loop that is causing the broken behaviour?

Another example to try in IDLE:

```
answer = input("What is the capital of New Zealand? ")
while answer != "Wellington":
    print("Sorry, that is not correct. Try again")
    answer = input("What is the capital of New Zealand? ")
```

Run this code several times. Test what happens if you put in the correct answer straight away, or if you put in some wrong answers before the correct one.

As with if statements and for loops, any code inside the while loop must be indented by 4 spaces. Try the previous examples out in IDLE.

Test Yourself:

Write some code to accomplish the following task in order:

1. Print each number from 0 to 100
2. Print the first 5 square numbers
3. Print all the even number less than 100
4. Ask the user a question, and continue asking them until they get it correct
5. Use a while loop to print out the terms of the Fibonacci sequence which are less than 1000. To find the next term of the fibonacci sequence, add the previous two terms. Here are the first several terms:

1

1

2 (1 + 1)

3 (1 + 2)

5 (2 + 3)

8 (3 + 5)

13 (5 + 8)

etc...